Noname manuscript No.
(will be inserted by the editor)

Path-Following Gradient-Based Decomposition Algorithms For Separable Convex Optimization

Quoc Tran Dinh · **Ion Necoara** · **Moritz Diehl**

Received: date / Accepted: date

Abstract A new decomposition optimization algorithm, called *path-following gradient-based decomposition*, is proposed to solve separable convex optimization problems. Unlike path-following Newton methods considered in the literature, this algorithm does not requires any smoothness assumption on the objective function. This allows us to handle more general classes of problems arising in many real applications than in the path-following Newton methods. The new algorithm is a combination of three techniques, namely smoothing, Lagrangian decomposition and path-following gradient framework. The algorithm decomposes the original problem into smaller subproblems by using dual decomposition and smoothing via self-concordant barriers, updates the dual variables using a path-following gradient method and allows one to solve the subproblem in parallel. Moreover, the algorithmic parameters are updated automatically without any tuning strategy as in augmented Lagrangian approaches. We prove the global convergence of the new algorithm and analyze its local convergence rate. Then, we modify the proposed algorithm by applying Nesterov's accelerating scheme to get a new variant which has a better local convergence rate. Finally, we present preliminary numerical tests that confirm the theory development.

Keywords Path-following gradient method · dual fast gradient algorithm · separable convex optimization · smoothing technique · self-concordant barrier · parallel implementation.

1 Introduction

Many optimization problems arising in engineering and economics can conveniently be formulated as *Separable Convex Programming Problems* (SepCPs). Particularly, optimization problems related to a network $\mathcal{N}(V,E)$ of M agents, where V denotes the set of nodes and

Q. Tran Dinh and M. Diehl
Optimization in Engineering Center (OPTEC) and Department of Electrical Engineering, Katholieke Universiteit Leuven, Belgium, E-mail: {quoc.trandinh, moritz.diehl}@esat.kuleuven.be

I. Necoara
Automation and Systems Engineering Department, University Politehnica Bucharest, 060042 Bucharest, Romania, E-mail: ion.necoara@acse.pub.ro

Q. Tran Dinh
Department of Mathematics-Mechanics-Informatics, Vietnam National University, Hanoi, Vietnam.

E denotes the set of edges in the network, can be cast into separable convex optimization problems. Several applications can be found in the literature such as distributed control, network utility maximization, resource allocation, machine learning and multistage stochastic convex programming [1,2,17,21,22]. Problems of moderate size or possessing a sparse structure can be solved by standard optimization methods in a centralized setup. However, in many real applications we meet problems, which may not be suitable to solve by standard optimization approaches or exploiting problem structures, e.g. nonsmooth separate objective functions, dynamic structure or distributed information. In those situations, decomposition methods can be considered as an appropriate framework to tackle those problems. Particularly, Lagrangian dual decomposition techniques are widely used to decompose a large-scale separable convex optimization problem into smaller subproblem components, which can simultaneously be solved in a *parallel manner* or in a *closed form*.

Various approaches have been proposed to solve (SepCP) in decomposition framework. One class of algorithms is based on Lagrangian relaxation and subgradient-type methods of multipliers [1,5,12]. It has been observed that subgradient methods are usually slow and numerically sensitive to the choice of step sizes in practice [13]. The second approach relies on augmented Lagrangian functions, see e.g. [6,18]. Many variants were proposed to process the inseparability of the crossproduct terms in the augmented Lagrangian function in different ways. Another research direction is based on alternating direction methods which were studied, for example, in [2,7]. Alternatively, proximal point-type methods were extended to the decomposition framework, see, e.g. [3,10]. Other researchers employed interior point methods in the framework of decomposition such as [8,11,19,22].

In this paper, we follow the same line of the dual decomposition framework but in a different way. First, we smooth the dual function by using self-concordant barriers. By an appropriate choice of the smoothness parameter, we show that the dual function of the smoothed problem is an approximation of the original dual function. Then, we develop a new path-following gradient method for solving the smoothed dual problem. By strong duality, we can also recover an approximate solution for the original problem. Compared to the previous related methods mentioned above, the new approach has the following advantages. First, since a self-concordant barrier function only depends on its barrier parameter, this allows us to avoid a dependency on the diameter of the feasible set as in prox-function smoothing techniques [10,19]. Second, the proposed method is a gradient-type scheme which allows to handle more general classes of problems than in path-following Newton methods [9,19,22], in particular the nonsmoothness of the objective function. Third, by smoothing via self-concordant barrier functions, if the objective function is smooth then instead of solving the primal subproblems as general convex programs we can treat them by using optimality conditions which are equivalent to solving nonlinear systems. Finally, by convergence analysis, we provide an adaptive update for all the algorithmic parameters which still ensure the convergence of the new methods.

Contribution. The contribution of the paper can be summarized as follows:

(a) We propose to use a smoothing technique via barrier function to smooth the dual function of (SepCP) as in [8,9,22]. However, we provide a new estimate for the dual function, see Lemma 1.
(b) We propose a new path-following gradient-based decomposition algorithm, Algorithm 1, to solve (SepCP). This algorithm allows one to solve the subproblem of each component in parallel. Moreover, all the algorithmic parameters are updated automatically without using any tuning strategy.
(c) We prove the convergence of the algorithm and estimate its local convergence rate.

(d) We modify the algorithm by applying Nesterov's accelerating scheme to obtain a new variant, Algorithm 2, which possesses a convergence rate, i.e. $O(1/\varepsilon)$, where ε is a given accuracy.

Let us emphasize the following points. The new estimate of the dual function considered in this paper is different from the one in [19] which does not depend on the diameter of the feasible set of the dual problem. The worst case complexity of the second algorithm is $O(1/\varepsilon)$ which is much higher than in subgradient-type methods of multipliers [1,5,12]. We notice that this convergence rate is optimal in the sense of Nesterov's optimal schemes [13, 14] for this class of algorithms. Moreover, we can choose smoothness parameter to adjust this convergence rate. All the algorithms can be implemented in a *parallel manner*.

Outline. The rest of this paper is organized as follows. In the next section, we state the problem formulation and review the Lagrangian dual decomposition framework. Section 3 considers a smoothing technique via self-concordant barriers and provides an estimate for the dual function. The new algorithms and their convergence analysis are presented in Sections 4 and 5. Preliminarily numerical results are shown in the last section to verify our theoretical results.

Notation and Terminology. Throughout the paper, we work on the Euclidean space $\mathbb{R}^n$ endowed with an inner product x^Ty for $x,y \in \mathbb{R}^n$ and the norm $\|x\|_2 := \sqrt{x^Tx}$. For a proper, lower semi-continuous convex function f, $\partial f(x)$ denotes the subdifferential of f at x. If f is concave then we also use $\partial f(x)$ for its super-differential at x. For any $x \in \mathrm{dom}(f)$ such that $\nabla^2 f(x)$ is positive definite, the local norm of a vector u with respect to f at x is defined as $\|u\|_x := \left[u^T\nabla^2 f(x)u\right]^{1/2}$ and its dual norm is $\|u\|_x^* := \max\left\{u^Tv \mid \|v\|_x \le 1\right\} = \left[u^T\nabla^2 f(x)^{-1}u\right]^{1/2}$. It is obvious that $u^Tv \le \|u\|_x\|v\|_x^*$. The notation $\mathbb{R}_+$ and $\mathbb{R}_{++}$ define the sets of nonnegative and positive numbers, respectively. The function $\omega : \mathbb{R}_+ \to \mathbb{R}$ is defined by $\omega(t) := t - \ln(1+t)$ and its dual function $\omega_* : [0,1) \to \mathbb{R}$ is $\omega_*(t) := -t - \ln(1-t)$.

2 Separable Convex Programming Problems and Lagrangian Dual Decomposition

A *Separable Convex Programming* problem (SepCP) is typically written as follows:

$$\phi^* := \begin{cases} \max_x \ \phi(x) := \sum_{i=1}^N \phi_i(x_i), \\ \text{s.t.} \ \ \sum_{i=1}^N (A_ix_i - b_i) = 0, \\ \qquad x_i \in X_i, \ i = 1, \cdots, N, \end{cases} \tag{SepCP}$$

where the decision variable $x := (x_1, \ldots, x_N)$ with $x_i \in \mathbb{R}^{n_i}$, the function $\phi_i : \mathbb{R}^{n_i} \to \mathbb{R}$ is concave and the feasible set is described by the set $X := X_1 \times \cdots \times X_N$, with $X_i \in \mathbb{R}^{n_i}$ nonempty, closed, convex sets for all $i = 1, \cdots, N$. Matrix $A := [A_1, \ldots, A_N]$, with $A_i \in \mathbb{R}^{m\times n_i}$ for $i = 1, \ldots, N$, $b := \sum_{i=1}^N b_i \in \mathbb{R}^m$ and $n_1 + \cdots + n_N = n$. The constraint $Ax - b = 0$ in (SepCP) is called *coupling linear constraint*, while $x_i \in X_i$ is referred to as *local constraints* of the i-th component (agent).

Let $\mathscr{L}(x,y) := \phi(x) + y^T(Ax - b)$ be the partial Lagrangian function associated with the coupling constraint $Ax - b = 0$ of (SepCP). The dual problem of (SepCP) is written as:

$$g^* := \min_{y\in\mathbb{R}^m} g(y), \tag{1}$$

where g is the dual function defined by:

$$g(y) := \max_{x\in X} \mathscr{L}(x,y) = \max_{x\in X} \left\{ \phi(x) + y^T(Ax-b) \right\}. \tag{2}$$

Due to the separability of ϕ, the dual function g can be computed *in parallel* as:

$$g(y) = \sum_{i=1}^{N} g_i(y), \quad g_i(y) := \max_{x_i\in X_i} \left\{ \phi_i(x_i) + y^T(A_i x_i - b_i) \right\}, \ i = 1,\dots,N. \tag{3}$$

Throughout this paper, we make the following assumptions:

Assumption A.1 *The following assumptions hold, see [18]:*

(a) *The solution set X^* of* (SepCP) *is nonempty.*
(b) *Either X is polyhedral or the following Slater qualification condition holds:*

$$\mathrm{ri}(X) \cap \{x \mid Ax - b = 0\} \neq \emptyset, \tag{4}$$

where $\mathrm{ri}(X)$ *is the relative interior of X.*
(c) *The functions ϕ_i, $i = 1,\dots,N$, are proper, upper semicontinuous and concave and A is full-row rank.*

Assumption **A.1** is standard in convex optimization. Under this assumption, *strong duality* holds, i.e. the dual problem (1) is also solvable and $g^* = \phi^*$. Moreover, the set of Lagrange multipliers, Y^*, is bounded. However, under Assumption **A.1**, the dual function g may not be differentiable. Numerical methods such as subgradient-type and bundle methods can be used to solve (1). Nevertheless, these methods are in general numerically intractable and slow.

3 Smoothing via self-concordant barrier functions

In many practical problems, the feasible sets X_i, $i = 1,\dots,N$ are usually simple, e.g. box, polyhedra and ball. Hence, X_i can be endowed with a *self-concordant barrier* (see, e.g. [16, 13]) as in the following assumption.

Assumption A.2 *Each feasible set X_i, $i = 1,\dots,N$, is bounded and endowed with a self-concordant barrier function F_i with the parameter $\nu_i > 0$.*

Note that the assumption on the boundedness of X_i can be removed by assuming that the set of sample points generated by the new algorithm described below is bounded.

Remark 1 The theory developed in this paper can be easily extended to the case X_i given as follows (see [15]) for some $i \in \{1,\cdots,N\}$:

$$X_i := X_i^c \cap X_i^a, \ \ X_i^a := \{x_i \in \mathbb{R}^{n_i} \ : \ D_i x_i = d_i\}, \tag{5}$$

by applying the standard linear algebra routines, where the set X_i^c has nonempty interior and associated with a ν_i-self-concordant barrier F_i.

Let us denote by x_i^c the analytic center of X_i, i.e.:

$$x_i^c := \arg\min_{x_i \in \mathrm{int}(X_i)} F_i(x_i) \ \forall i = 1,\dots,N, \tag{6}$$

where $\mathrm{int}(X_i)$ is the interior of X_i. Since X_i is bounded, x_i^c is well-defined [13]. Moreover, the following estimates hold:

$$F_i(x_i) - F_i(x_i^c) \geq \omega(\|x_i - x_i^c\|_{x_i^c}) \text{ and } \|x_i - x_i^c\|_{x_i^c} \leq \nu_i + 2\sqrt{\nu_i}, \ \ \forall x_i \in X_i, \ i = 1,\dots,N. \tag{7}$$

Without loss of generality, we can assume that $F_i(x_i^c) = 0$. Otherwise, we can replace F_i by $\tilde{F}_i(\cdot) := F_i(\cdot) - F_i(x_i^c)$ for $i = 1,\dots,N$. Since X is separable, $F := \sum_{i=1}^N F_i$ is a self-concordant barrier of X with the parameter $\nu := \sum_{i=1}^N \nu_i$.

Let us define the following function::

$$g(y;t) := \sum_{i=1}^N g_i(y;t), \tag{8}$$

where

$$g_i(y;t) := \max_{x_i \in \mathrm{int}(X_i)} \left\{ \phi_i(x_i) + y^T(A_i x_i - b_i) - tF_i(x_i) \right\}, \ i = 1,\dots,N, \tag{9}$$

with $t > 0$ being referred to as a smoothness parameter. Note that the maximum problem in (9) has a unique optimal solution, which is denoted by $x_i^*(y;t)$, due to the strict concavity of the objective function. We call this problem the *primal subproblem*. Consequently, the functions $g_i(\cdot,t)$ and $g(\cdot,t)$ are well-defined and smooth on $\mathbb{R}^m$ for any $t > 0$. We call $g_i(\cdot;t)$ and $g(\cdot;t)$ the *smoothed dual function* of g_i and g, respectively.

The optimality condition for (9) is:

$$0 \in \partial\phi_i(x_i^*(y;t)) + A_i^T y - t\nabla F_i(x_i^*(y;t)), \ i = 1,\cdots,N. \tag{10}$$

Let us define the full optimal solution $x^*(y;t) := (x_1^*(y;t),\cdots,x_N^*(y;t))$. Since problem (9) is convex, this condition (10) is necessary and sufficient for optimality. Moreover, the gradients of $g_i(\cdot;t)$ and $g(\cdot;t)$ are given by:

$$\nabla g_i(y;t) = A_i x_i^*(y;t) - b_i, \ \ \nabla g(y;t) = Ax^*(y;t) - b. \tag{11}$$

If f_i is differentiable for some $i \in \{1,\cdots,N\}$ then the condition (10) collapses to $\nabla\phi_i(x_i^*(y;t)) + A_i^T y - t\nabla F_i(x_i^*(y;t)) = 0$, which is indeed a *system of nonlinear equations*. First, we prove that $g(\cdot;t)$ is an approximation of the dual function $g(\cdot)$ for sufficiently small $t > 0$.

Lemma 1 *Suppose that Assumptions **A.1** and **A.2** are satisfied. Let $\bar{x}$ be a strictly feasible point to* (SepCP)*, i.e. $\bar{x} \in \mathrm{int}(X) \cap \{x \mid Ax = b\}$. Then, for any $t > 0$, we have:*

$$g(y) - \phi(\bar{x}) \geq 0 \ \ \textit{and} \ \ g(y;t) - \phi(\bar{x}) + tF(\bar{x}) \geq 0. \tag{12}$$

Moreover, it holds that:

$$g(y;t) \leq g(y) \leq g(y;t) + t(\nu + F(\bar{x})) + 2\sqrt{t\nu}\,[g(y;t) + tF(\bar{x}) - \phi(\bar{x})]^{1/2}. \tag{13}$$

Proof The first two inequalities in (12) are trivial due to the definitions of $g(\cdot)$, $g(\cdot;t)$ and the feasibility of $\bar{x}$. We only prove (13). Indeed, since $\bar{x} \in \mathrm{int}(X)$ and $x^*(y) \in X$, if we define $x^*_\tau(y) := \bar{x} + \tau(x^*(y) - \bar{x})$, then $x_\tau(y) \in \mathrm{int}X$ if $\tau \in [0,1)$. By applying the inequality [16, 2.3.3] we have:

$$F(x_\tau(y)) \le F(\bar{x}) - \nu\ln(1-\tau).$$

Using this inequality together with the definition of $g(\cdot;t)$, the concavity of ϕ and $A\bar{x} = b$, we deduce:

$$\begin{aligned} g(y;t) &= \max_{x\in\mathrm{int}(X)} \left\{\phi(x) + y^T(Ax-b) - tF(x)\right\} \\ &\ge \max_{\tau\in[0,1)} \left\{\phi(x_\tau(y)) + y^T(Ax_\tau(y)-b) - tF(x_\tau(y))\right\} \\ &\ge \max_{\tau\in[0,1)} \{(1-\tau)\phi(\bar{x}) + \tau g(y) + t\nu\ln(1-\tau)\} - tF(\bar{x}). \end{aligned} \tag{14}$$

By solving the maximization problem on the right hand side of (14) and then rearranging the results, we obtain:

$$g(y) \le g(y;t) + t[\nu + F(\bar{x})] + t\nu\left[\ln\left(\frac{g(y)-\phi(\bar{x})}{t\nu}\right)\right]_+, \tag{15}$$

where $[\cdot]_+ := \max\{\cdot, 0\}$. Moreover, it follows from (14) that:

$$\begin{aligned} g(y) - \phi(\bar{x}) &\le \frac{1}{\tau}\left[g(y;t) - \phi(\bar{x}) + tF(\bar{x}) + t\nu\ln(1 + \frac{\tau}{1-\tau})\right] \\ &\le \frac{1}{\tau}\left[g(y;t) - \phi(\bar{x}) + tF(\bar{x})\right] + \frac{t\nu}{1-\tau}. \end{aligned}$$

If we minimize the right hand side of this inequality in $[0,1)$, then we get $g(y) - \phi(\bar{x}) \le [(g(y;t) - \phi(\bar{x}) + tF(\bar{x}))^{1/2} + \sqrt{t\nu}]^2$. Finally, we plug this inequality into (15) to obtain:

$$\begin{aligned} g(y) &\le g(y;t) + t\nu + 2t\nu\ln\left(1 + \sqrt{\frac{[g(y;t) - \phi(\bar{x}) + tF(\bar{x}]}{t\nu}}\right) + tF(\bar{x}) \\ &\le g(y;t) + t\nu + tF(\bar{x}) + 2\sqrt{t\nu}\,[g(y;t) - \phi(\bar{x}) + tF(\bar{x})]^{1/2}, \end{aligned}$$

which is indeed (13). □

Remark 2 (Approximation of $g(y)$) It follows from (13) that $g(y) \le (1 + 2\sqrt{t\nu})g(y;t) + t(\nu + F(\bar{x})) + 2\sqrt{t\nu}(tF(\bar{x}) - \phi(\bar{x}))$. Hence, $g(y;t) \to g(y)$ as $t \to 0^+$. Moreover, this estimate is different from the one in [19], since we do not assume that Y is bounded.

Next, we consider the following minimization problem, called *smoothed dual problem*:

$$g^*(t) := g(y^*(t);t) = \min_{y\in\mathbb{R}^m} g(y;t). \tag{16}$$

We denote by $y^*(t)$ the solution of (16). The following lemma shows the main properties of the functions $g(y;\cdot)$ and $g^*(\cdot)$.

Lemma 2 *Suppose that Assumptions **A.1** and **A.2** are satisfied. Then:*

(a) *The function $g(y;\cdot)$ is convex and nonincreasing in $\mathbb{R}_{++}$ for a given $y \in \mathbb{R}^m$. Moreover, we have:*

$$g(y;\hat{t}) \ge g(y;t) - (\hat{t} - t)F(x^*(y;t)). \tag{17}$$

(b) *The function* $g^*(\cdot)$ *defined by* (16) *is differentiable and nonincreasing in* $\mathbb{R}_{++}$. *Moreover,* $g^*(t) \le g^*$, $\lim_{t\downarrow 0^+} g^*(t) = g^* = \phi^*$ *and* $x^*(y^*(t);t)$ *is feasible to the original problem* (SepCP).

Proof We only prove (17), the proof of the remainders can be found in [9,19]. Indeed, since $g(y;\cdot)$ is convex and differentiable and $\frac{dg(y;t)}{dt} = -F(x^*(y;t)) \le 0$, we have $g(y;\hat{t}) \ge g(y;t) + (\hat{t}-t)\frac{dg(y;t)}{dt} = g(y;t) - (\hat{t}-t)F(x^*(y;t))$. □

The statement (b) of Lemma 2 shows that if we find an approximate solution y^k for (16) for sufficiently small t_k, then $g^*(t_k)$ approximates g^* (recall that $g^* = \phi^*$) and $x^*(y^k;t_k)$ is approximately feasible to (SepCP).

4 Path-following gradient method

In this section we design a path-following gradient algorithm to solve the dual problem (1), analyze the convergence of the algorithm and estimate the local convergence rate.

4.1 The path-following gradient scheme

Since $g(\cdot;t)$ is strictly convex and smooth, we can write the optimality condition of (16) as:

$$\nabla g(y;t) = 0. \tag{18}$$

This equation has a unique solution $y^*(t)$.

Now, for any given $x \in \operatorname{int}(X)$, $\nabla F(x)$ is positive definite. We introduce a local matrix norm:

$$|||A|||_x^* := \|A\nabla^2 F(x)^{-1}A^T\|_2, \tag{19}$$

The following lemma shows a main property of the function $g(\cdot;t)$.

Lemma 3 *Suppose that Assumptions* ***A.1*** *and* ***A.2*** *are satisfied. Then, for all* $t > 0$ *and* $y, \hat{y} \in \mathbb{R}^m$, *one has:*

$$[\nabla g(y;t) - \nabla g(\hat{y};t)]^T(y-\hat{y}) \ge \frac{t\|\nabla g(y;t) - \nabla g(\hat{y};t)\|_2^2}{c_A\left[c_A + \|\nabla g(y,t) - \nabla g(\hat{y};t)\|_2\right]}, \tag{20}$$

where $c_A := |||A|||^*_{x^*(y;t)}$. *Consequently, it holds that:*

$$g(\hat{y};t) \le g(y;t) + \nabla g(y;t)^T(\hat{y}-y) + t\omega^*(c_A t^{-1}\|\hat{y}-y\|_2), \tag{21}$$

provided that $c_A\|\hat{y}-y\|_2 < t$.

Proof For notational simplicity, we denote by $x^* := x^*(y;t)$ and $\hat{x}^* := x^*(\hat{y};t)$. From the definition (11) of $\nabla g(\cdot;t)$ and the Cauchy-Schwarz inequality we have:

$$[\nabla g(y;t) - \nabla g(\hat{y};t)]^T(y-\hat{y}) = (y-\hat{y})^T A(x^* - \hat{x}^*). \tag{22}$$

$$\|\nabla g(\hat{y};t) - \nabla g(y;t)\|_2 \le |||A|||^*_{x^*}\|\hat{x}^* - x^*\|_{x^*}. \tag{23}$$

It follows from (10) that $A^T(y-\hat{y}) = t[\nabla F(x^*) - \nabla F(\hat{x}^*] - [\xi(x^*) - \xi(\hat{x}^*)]$, where $\xi(\cdot) \in \partial\phi(\cdot)$. By multiplying this relation with $x^* - \hat{x}^*$ and then using [13, Theorem 4.1.7] and the concavity of ϕ we obtain:

$$\begin{aligned}(y-\hat{y})^TA(x^*-\hat{x}^*) &= t[\nabla F(x^*) - \nabla F(\hat{x}^*)]^T(x^*-\hat{x}^*) - [\xi(x^*) - \xi(\hat{x}^*)]^T(x^*-\hat{x}^*)\\ &\overset{\text{concavity of }\phi}{\geq} t[\nabla F(x^*) - \nabla F(\hat{x}^*)]^T(x^*-\hat{x}^*)\\ &\geq \frac{t\|x^*-\hat{x}^*\|_{x^*}^2}{1+\|x^*-\hat{x}^*\|_{x^*}}\\ &\overset{(23)}{\geq} \frac{t\,[\|\nabla g(y;t) - \nabla g(\hat{y};t)\|_2]^2}{|||A|||_{x^*}^*\left[|||A|||_{x^*}^* + \|\nabla g(y;t) - \nabla g(\hat{y};t)\|_2\right]}.\end{aligned}$$

Substituting this inequality into (22) we obtain (20).

By the Cauchy-Schwarz inequality, it follows from (20) that $\|\nabla g(\hat{y};t) - \nabla g(y;t)\| \leq \frac{c_A^2\|\hat{y}-y\|_2}{t - c_A\|\hat{y}-y\|}$, provided that $c_A\|\hat{y}-y\| \leq t$. Finally, by using the mean-value theorem, we have:

$$\begin{aligned}g(\hat{y};t) &= g(y;t) + \nabla g(y;t)^T(\hat{y}-y) + \int_0^1 (\nabla g(y+s(\hat{y}-y);t) - \nabla g(y;t))^T(\hat{y}-y)ds\\ &\leq g(y;t) + \nabla g(y;t)^T(\hat{y}-y) + c_A\|\hat{y}-y\|_2 \int_0^1 \frac{c_As\|\hat{y}-y\|_2}{t - c_As\|\hat{y}-y\|_2}ds\\ &= g(y;t) + \nabla g(y;t)^T(\hat{y}-y) + t\omega^*(c_At^{-1}\|\hat{y}-y\|_2),\end{aligned}$$

which is indeed (21) provided that $c_A\|\hat{y}-y\|_2 < t$. □

Now, we describe one step of the path-following gradient method for solving (16). Let us assume that $y \in \mathbb{R}^m$ and $t > 0$ are the values at the current iteration, the values y_+ and t_+ at the next iteration are computed as:

$$\begin{cases} t_+ := t - \Delta t,\\ y_+ := y - \alpha\nabla g(y, t_+),\end{cases} \tag{24}$$

where $\alpha := \alpha(y;t) > 0$ is the current step size and Δt is the decrement of the parameter t. In order to analyze the convergence of the scheme (24), we introduce the following notation:

$$x_1^* := x^*(y;t_+),\ \ c_{A1} = |||A|||_{x^*(y;t_+)}^* \text{ and } \lambda_1 := \|\nabla g(y;t_+)\|_2. \tag{25}$$

First, we prove an important property of the *path-following gradient scheme* (24).

Lemma 4 *Under Assumptions **A.1** and **A.2**, the following inequality holds:*

$$g(y_+;t_+) \leq g(y;t) - \left[\alpha\lambda_1^2 - t_+\omega^*(c_{A1}t_+^{-1}\alpha\lambda_1) - \Delta tF(x_1^*)\right], \tag{26}$$

where c_{A1} and λ_1 are defined by (25).

Proof Since $t_+ = t - \Delta t$, by using (17) with t and t_+, we have:

$$g(y;t_+) \leq g(y;t) + \Delta tF(x^*(y;t_+)). \tag{27}$$

Next, by (21) we have $y_+ - y = -\alpha\nabla g(y;t_+)$ and $\lambda_1 := \|\nabla g(y;t_+)\|_2$. Hence, we can derive:

$$g(y_+;t_+) \leq g(y;t_+) - \alpha\lambda_1^2 + t_+\omega^*(c_{A1}\alpha\lambda_1t_+^{-1}). \tag{28}$$

By plugging (27) into (28), we obtain (26). □

Lemma 5 *For any $y \in \mathbb{R}^m$ and $t > 0$, the constant $c_A := |||A|||^*_{x^*(y;t_+)}$ is bounded. More precisely, $c_A \leq \bar{c}_A := \kappa |||A|||^*_{x^c} < +\infty$. Furthermore, $\lambda := \|\nabla g(y;t)\|_2$ is also bounded, i.e.: $\lambda \leq \bar{\lambda} := \kappa |||A|||^*_{x^c} + \|Ax^c - b\|_2$, where $\kappa := \sum_{i=1}^N [\nu_i + 2\sqrt{\nu_i}]$.*

Proof For any $x \in \mathrm{int}(X)$, from the definition of $|||\cdot|||^*_x$, we have:

$$\begin{aligned}
|||A|||^*_x &= \sup\left\{[v^T A \nabla^2 F(x)^{-1} A^T v]^{1/2} \ : \ \|v\|_2 = 1\right\}\\
&= \sup\left\{\|u\|^*_x \ : \ u = A^T v, \ \|v\|_2 = 1\right\}\\
&\leq \sup\left\{(\nu + 2\sqrt{\nu})\|u\|^*_{x^c} \ : \ u = A^T v, \ \|v\|_2 = 1\right\}\\
&= (\nu + 2\sqrt{\nu}) \sup\left\{\left[v^T A \nabla^2 F(x^c)^{-1} A^T v\right]^{1/2}, \ \|v\|_2 = 1\right\}\\
&= (\nu + 2\sqrt{\nu}) |||A|||^*_{x^c}.
\end{aligned}$$

Here, the inequality in this implication follows from [13, Corollary 4.2.1]. By substituting $x = x^*(y;t)$ into the above inequality, we obtain the first conclusion. In order to prove the second bound, we note that $\nabla g(y;t) = Ax^*(y;t) - b$. Therefore, by using (7), we can estimate:

$$\begin{aligned}
\|\nabla g(y;t)\|_2 &= \|Ax^*(y;t) - b\|_2 \leq \|A(x^*(y;t) - x^c)\|_2 + \|Ax^c - b\|_2\\
&\leq |||A|||^*_{x^c} \|x^*(y;t) - x^c\|_{x^c} + \|Ax^c - b\|_2\\
&\overset{(7)}{\leq} \kappa |||A|||^*_{x^c} + \|Ax^c - b\|_2,
\end{aligned}$$

which is the second conclusion. □

Next, we show how to choose the step size α and the decrement Δt such that $g(y_+;t_+) < g(y;t)$ in Lemma 4. We note that $x^*(y;t_+)$ is obtained by solving the subproblem (9) and the quantity $c_F := F(x^*(y;t_+))$ is nonnegative and computable. By Lemma 5, we see that:

$$\alpha(t) := \frac{t}{c_{A1}(c_{A1} + \lambda_1)} \geq \underline{\alpha}_0(t) := \frac{t}{\bar{c}_A(\bar{c}_A + \bar{\lambda})}, \tag{29}$$

which shows that $\alpha(t)$ is bounded away from zero. We have the following estimate.

Lemma 6 *The step size $\alpha(t)$ defined by* (29) *satisfies:*

$$g(y_+;t_+) \leq g(y;t) - t_+ \omega\left(\frac{\lambda_1}{c_{A1}}\right) + \Delta t F(x_1^*). \tag{30}$$

Proof Let $\varphi(\alpha) := \alpha\lambda_1^2 - t_+\omega^*(c_{A1}t_+^{-1}\alpha\lambda_1) - t_+\omega(\lambda_1 c_{A1}^{-1})$. We can simplify this function as $\varphi(\alpha) = t_+[u + \ln(1-u)]$, where $u := t_+^{-1}\lambda_1^2\alpha + t_+^{-1}c_{A1}\lambda_1\alpha - c_{A1}^{-1}\lambda_1$. The function $\varphi(\alpha) \leq 0$ for all u and $\varphi(\alpha) = 0$ at $u = 0$ which leads to $\alpha := \frac{t}{c_{A1}(c_{A1}+\lambda_1)}$. □

Since $t_+ = t - \Delta t$, if we choose $\Delta t := \frac{t\omega(c_{A1}^{-1}\lambda_1)}{2[\omega(c_{A1}^{-1}\lambda_1) + F(x_1^*)]}$ then:

$$g(y_+;t_+) \leq g(y;t) - \frac{t}{2}\omega(c_{A1}^{-1}\lambda_1). \tag{31}$$

Therefore, the update rule for t can be written as:

$$t_+ := (1-\sigma)t, \text{ where } \sigma := \frac{\omega(c_{A1}^{-1}\lambda_1)}{2[\omega(c_{A1}^{-1}\lambda_1) + F(x_1^*)]} \in (0,1). \tag{32}$$

4.2 The algorithm

Combing the above analysis, we can describe the path-following gradient decomposition method is follows:

Algorithm 1. (*Path-following gradient decomposition*).

Initialization:

1. Choose an initial value $t_0 > 0$ and tolerances $\varepsilon_t > 0$ and $\varepsilon_g > 0$.
2. Take an initial point $y^0 \in \mathbb{R}^m$ and solve (3) *in parallel* to obtain $x_0^* := x^*(y^0; t_0)$.
3. Compute $c_A^0 := |||A|||_{x_0^*}^*$, $\lambda_0 := \|\nabla g(y^0; t_0)\|_2$, $\omega_0 := \omega(\lambda_0 / c_A^0)$ and $c_F^0 := F(x_0^*)$.

Iteration: For $k = 0, 1, \cdots$, perform the following steps:

Step 1: Update the barrier parameter: $t_{k+1} := t_k(1 - \sigma_k)$, where $\sigma_k := \frac{\omega_k}{2(\omega_k + c_F^k)}$.

Step 2: Solve (3) *in parallel* to obtain $x_k^* := x^*(y^k, t_{k+1})$. Then, form the gradient vector $\nabla g(y^k; t_{k+1}) := Ax_k^* - b$.

Step 3: Compute $\lambda_{k+1} := \|\nabla g(y^k; t_{k+1})\|_2$, $c_A^{k+1} := |||A|||_{x_k^*}^*$, $\omega_{k+1} := \omega(\lambda_{k+1} / c_A^{k+1})$ and $c_F^{k+1} := F(x_k^*)$.

Step 4: If $t \le \varepsilon_t$ and $\lambda_k \le \varepsilon$, then terminate.

Step 5: Compute the step size $\alpha_{k+1} := \frac{t_{k+1}}{c_A^{k+1}(c_A^{k+1} + \lambda_{k+1})}$.

Step 6: Update y^{k+1} as:

$$y^{k+1} := y^k - \alpha_{k+1} \nabla g(y^k, t_{k+1}).$$

End.

The main step of Algorithm 1 is Step 2, where we need to solve in parallel the primal subproblems. To form the gradient vector $\nabla g(\cdot, t_{k+1})$, one can compute in parallel by multiplying column-blocks A_i of A by the solution $x_i^*(y^k, t_{k+1})$. This task only requires local information to be exchanged between the current node and its neighbors.

From the update rule (32) of t we can see that $\sigma_k \to 0^+$ as $F(x_k^*) \to \infty$. This happens when the barrier function is approaching the boundary of the feasible set X. Hence, the parameter t is not decreased. Let $\bar{c}_F$ be a sufficiently large positive constant. We can modify the update rule of t as:

$$t_{k+1} := \begin{cases} t_k\left(1 - \frac{\omega_k}{2(\omega_k + c_F^k)}\right) & \text{if } c_F^k \le \bar{c}_F, \\ t_k & \text{otherwise,} \end{cases} \tag{33}$$

In this case, the sequence $\{t_k\}$ generated by Algorithm 1 might not converge to zero. Moreover, the step size α_k computed at Step 5 depends on the parameter t_k. If t_k is small then Algorithm 1 makes short steps toward a solution of (1).

4.3 Convergence analysis

Let us assume that $\underline{t} = \inf_{k \ge 0} t_k > 0$. Then, the following theorem shows the convergence of Algorithm 1.

Theorem 1 *Suppose that Assumptions **A.1** and **A.2** are satisfied. Suppose further that the sequence* $\{(y^k, t_k, \lambda_k)\}_{k \ge 0}$ *generated by Algorithm 1 satisfies* $\underline{t} := \inf_{k \ge 0}\{t_k\} > 0$. *Then:*

$$\lim_{k \to \infty} \|\nabla g(y^k, t_{k+1})\|_2 = 0. \tag{34}$$

Consequently, there exists a limit point y^* *of* $\{y^k\}$ *such that* y^* *is a solution of* (16) *at* $t = \underline{t}$.

Proof It is sufficient to prove (34). Indeed, from (31) we have:

$$\sum_{i=0}^{k} \frac{t_k}{2}\omega(\lambda_{k+1}/c_A^{k+1}) \le g(y^0;t_0) - g(y^{k+1};t_{k+1}) \le g(y^0;t_0) - g^*.$$

Since $t_k \ge \underline{t} > 0$ and $c_A^{k+1} \le \bar{c}_A$ due to Lemma 5, the above inequality leads to:

$$\frac{\underline{t}}{2}\sum_{i=0}^{\infty}\omega(\lambda_{k+1}/\bar{c}_A) \le g(y^0;t_0) - g^* < +\infty.$$

This inequality implies $\lim_{k\to\infty}\omega(\lambda_{k+1}/\bar{c}_A) = 0$, which leads to $\lim_{k\to\infty}\lambda_{k+1} = 0$. By definition of λ_k we have $\lim_{k\to\infty}\|\nabla g(y^k;t_{k+1})\|_2 = 0$. $\square$

Remark 3 From the proof of Theorem 1, we can fix $c_A^k \equiv \bar{c} := \kappa|||A|||^*_{x^c}$ in Algorithm 1. This value can be computed a priori.

4.4 Local convergence rate

Let us analyze the local convergence rate of Algorithm 1. Let y^0 be an initial point of Algorithm 1 and $y^*(t)$ be the unique solution of (16). We denote by:

$$r_0(t) := \|y^0 - y^*(t)\|_2. \tag{35}$$

For simplicity of discussion, we assume that the smoothness parameter t_k is fixed at $\underline{t} > 0$ sufficiently small for all $k \ge 0$ (see Lemma 1). The convergence rate of Algorithm 1 in the case $t_k = \underline{t}$ is stated in the following lemma.

Lemma 7 (***Local convergence rate***) *Suppose that the initial point y^0 is chosen such that $g(y^0;\underline{t}) - g^*(\underline{t}) \le \frac{3\bar{c}_A}{2r_0(\underline{t})}$. Then:*

$$g(y^k;\underline{t}) - g^*(\underline{t}) \le \frac{12\bar{c}_A^2 r_0(\underline{t})^2}{16\bar{c}_A r_0(\underline{t}) + 3\underline{t}k}. \tag{36}$$

Consequently, the local convergence rate of Algorithm 1 is at least $O\left(\frac{4\bar{c}_A^2 r_0(\underline{t})^2}{\underline{t}k}\right)$.

Proof Let $\Delta_k := g(y^k;\underline{t}) - g^*(\underline{t})$ and $\underline{y}^* := y^*(\underline{t})$. Then $\Delta_k \ge 0$. First, by the convexity of $g(\cdot;\underline{t})$ we have:

$$\Delta_k = g(y^k;\underline{t}) - g^*(\underline{t}) \le \|\nabla g(y^k,\underline{t})\|_2\|y^k - \underline{y}^*\|_2 = \underline{\lambda}_k\|y^0 - \underline{y}^*\|_2 \le \underline{\lambda}_k r_0(\underline{t}).$$

This inequality implies:

$$\underline{\lambda}_k \ge r_0(\underline{t})^{-1}\Delta_k. \tag{37}$$

Since $t_k = \underline{t} > 0$ is fixed for all $k \ge 0$, it follows from (26) that:

$$g(y^{k+1};\underline{t}) \le g(y^k;\underline{t}) - \underline{t}\omega(\underline{\lambda}_k/\underline{c}_A^k),$$

where $\underline{\lambda}_k := \|\nabla g(y^k;\underline{t})\|_2$ and $\underline{c}_A^k := |||A|||^*_{x^*(y^k;\underline{t})}$. By using the definition of Δ_k, the last inequality is equivalent to:

$$\Delta_{k+1} \le \Delta_k - \underline{t}\omega(\underline{\lambda}_k/\underline{c}_A^k). \tag{38}$$

Next, since $\omega(\tau) \geq \tau^2/2 - \tau^3/3 \geq \tau^2/4$ for all $0 \leq \tau \leq 3/4$ and $\underline{c}_A^k \leq \bar{c}_A$ due to Lemma 5, it follows from (37) and (38) that:

$$\Delta_{k+1} \leq \Delta_k - (\underline{t}\Delta_k^2)/(4r_0(\underline{t})^2\bar{c}_A^2), \tag{39}$$

for all $\Delta_k \leq 3\bar{c}_A r_0(\underline{t})/4$.

Let $\eta := \underline{t}/(4r_0(\underline{t})^2\bar{c}_A^2)$. Since $\Delta_k \geq 0$, (39) implies:

$$\frac{1}{\Delta_{k+1}} \geq \frac{1}{\Delta_k(1-\eta\Delta_k)} = \frac{1}{\Delta_k} + \frac{\eta}{(1-\eta\Delta_k)} \geq \frac{1}{\Delta_k} + \eta.$$

By induction, this inequality leads to $\frac{1}{\Delta_k} \geq \frac{1}{\Delta_0} + \eta k$ which is equivalent to $\Delta_k \leq \frac{\Delta_0}{1+\eta\Delta_0 k}$ provided that $\Delta_0 \leq 3\bar{c}_A r_0(\underline{t})/4$. Since $\eta := \underline{t}/(4r_0(\underline{t})^2\bar{c}_A^2)$, this inequality is indeed (36). The last conclusion follows from (36). □

5 Fast gradient decomposition algorithm

Let us fix $t = \underline{t} > 0$. The function $g(\cdot;\underline{t})$ is convex and differentiable but its gradient is not Lipschitz continuous, we can not apply Nesterov's fast gradient algorithm [13] to solve (16). In this section, we modify Nesterov's fast gradient method in order to obtain an accelerating gradient method for solving (16).

One step of the modified fast gradient method is described as follows. Let y and v be given points in $\in \mathbb{R}^m$, we compute new points y_+ and v_+ as follows:

$$\begin{cases} y_+ := v - \alpha\nabla g(v;\underline{t}), \\ v_+ = \beta_1 y_+ + \beta_2 y + \beta_3 v, \end{cases} \tag{40}$$

where $\alpha := \underline{t}/(\bar{c}_A(\bar{c}_A+\lambda))$ is the step size, β_1, β_2 and β_3 are three parameters which will be chosen appropriately. First, we prove the following estimate.

Lemma 8 *Let $\theta \in (0,1)$ be a given parameter and $\rho := t/(2\theta\bar{c}_A^2)$. We define two vectors:*

$$r := \theta^{-1}[v - (1-\theta)y] \ \text{ and } \ r_+ = r - \rho\nabla g(v;\underline{t}). \tag{41}$$

Then the new point y_+ generated by (40) *satisfies:*

$$\theta^{-2}\big(g(y_+;\underline{t}) - \underline{g}^*\big) + \underline{t}^{-1}\bar{c}_A^2\|r_+ - \underline{y}^*\|_2^2 \leq \theta^{-2}(1-\theta)\big(g(y;\underline{t}) - \underline{g}^*\big) + \underline{t}^{-1}\bar{c}_A^2\|r - \underline{y}^*\|_2^2, \tag{42}$$

provided that $\|\nabla g(v;\underline{t})\|_2 \leq 3\bar{c}_A/4$, where $\underline{y}^* := y^*(\underline{t})$ and $\underline{g}^* := g(\underline{y}^*;\underline{t})$.

Proof Since $y_+ = v - \alpha\nabla g(v;\underline{t})$ and $\alpha = \frac{\underline{t}}{\bar{c}_A(\bar{c}_A+\lambda)}$, it follows from (21) that:

$$g(y_+;\underline{t}) \leq g(v;\underline{t}) - \underline{t}\omega(\|\nabla g(v;\underline{t})\|_2/\bar{c}_A). \tag{43}$$

Now, since $\omega(\tau) \geq \tau^2/4$ for all $0 \leq \tau \leq 3/4$, the inequality (43) implies:

$$g(y_+;\underline{t}) \leq g(v;\underline{t}) - \frac{\underline{t}}{4\bar{c}_A^2}\|\nabla g(v;\underline{t})\|_2^2, \tag{44}$$

provided that $\|\nabla g(v;\underline{t})\|_2 \leq 3\bar{c}_A/4$. For any $u := (1-\theta)y + \theta \underline{y}^*$ and $\theta \in (0,1)$ we have:

$$
\begin{aligned}
g(v;\underline{t}) &\leq g(u;\underline{t}) + \nabla g(v;\underline{t})^T(v-u) \leq (1-\theta)g(y;\underline{t}) + \theta g(\underline{y}^*;\underline{t})\\
&\quad + \nabla g(v;\underline{t})^T(v-(1-\theta)y - \theta \underline{y}^*).
\end{aligned} \tag{45}
$$

By substituting (45) and the relation $v-(1-\theta)y = \theta r$ into (44) we obtain:

$$
\begin{aligned}
g(y_+;\underline{t}) &\leq (1-\theta)g(y;\underline{t}) + \theta \underline{g}^* + \theta \nabla g(v;\underline{t})^T(r-\underline{y}^*) - \frac{\underline{t}}{4\bar{c}_A^2}\|\nabla g(v;\underline{t})\|_2^2\\
&= (1-\theta)g(y;\underline{t}) + \theta \underline{g}^* + \frac{\theta^2 \bar{c}_A^2}{\underline{t}}\left[\|r-\underline{y}^*\|_2^2 - \|r - \frac{\underline{t}}{2\theta \bar{c}_A^2}\nabla g(v;\underline{t}) - \underline{y}^*\|_2^2\right]\\
&= (1-\theta)g(y;\underline{t}) + \theta \underline{g}^* + \frac{\theta^2 \bar{c}_A^2}{\underline{t}}\left[\|r-\underline{y}^*\|_2^2 - \|r_+ - \underline{y}^*\|_2^2\right].
\end{aligned} \tag{46}
$$

Since $1/\theta^2 = (1-\theta)/\theta^2 + 1/\theta$, by rearranging (46) we obtain (42). □

Next, we consider the update rule of θ. We can see from (42) that if θ_+ is updated such that $(1-\theta_+)/\theta_+^2 = 1/\theta^2$ then $g(y_+;\underline{t}) < g(y;\underline{t})$. The above condition implies:

$$
\theta_+ = 0.5\theta(\sqrt{\theta^2+4} - \theta).
$$

The following lemma provides an estimate for $k \geq 0$.

Lemma 9 *The sequence $\{\theta_k\}_{k\geq 0}$ generated by $\theta_{k+1} := 0.5\theta_k[(\theta_k^2+4)^{1/2} - \theta_k]$ and $\theta_0 = 1$ satisfies:*

$$
\frac{1}{2k+1} \leq \theta_k \leq \frac{2}{k+2}, \quad \forall k \geq 0. \tag{47}
$$

Proof We note that $\theta_{k+1} = \frac{2}{\sqrt{1+4/\theta^2}+1}$. If we define $s_k := 2/\theta_k$ then the last relation implies $\frac{2}{s_{k+1}} = \frac{2}{\sqrt{s_k^2+1}+1}$, which leads to $\frac{2}{s_k+2} < \frac{2}{s_{k+1}} < \frac{2}{s_k+1}$. Hence, $s_k + 1 < s_{k+1} < s_k + 2$. By induction, we have $s_0 + k < s_k < s_0 + 2k$. More over, $\theta_0 = 1$, we have $s_0 = 2$. Substituting $s_0 = 2$ into the last inequalities and then using the relation $s_k = 2/\theta_k$ we obtain (47). □

By Lemma 8, we have $r_+ = r - \rho \nabla g(v;\underline{t})$ and $r_+ = \theta_+^{-1}(v_+ - (1-\theta_+)y_+)$. From these relations, we deduce that:

$$
v_+ = (1-\theta_+)y_+ + \theta_+(r - \rho \nabla g(v;\underline{t})). \tag{48}
$$

Note that if we combine (48) and (40) then:

$$
v_+ = (1-\theta_+ - \alpha^{-1}\rho\theta_+)y_+ - \theta^{-1}\theta_+(1-\theta)y + \left(\theta^{-1} + \alpha^{-1}\rho\right)\theta_+ v.
$$

This is the second line of (40), where $\beta_1 := 1 - \theta_+ - \rho\theta_+\alpha^{-1}$, $\beta_2 := -(1-\theta)\theta_+\theta^{-1}$ and $\beta_3 := (\theta^{-1} + \rho\alpha^{-1})\theta_+$. By combining all the above analysis, we can describe the modified fast gradient algorithm in detail as follows:
Algorithm 2. (*Fast gradient decomposition algorithm*).
Initialization: Perform the following steps:

1. Given a tolerance $\varepsilon > 0$. Fix the parameter t at a certain value $\underline{t} > 0$.
2. Find an initial point $y^0 \in \mathbb{R}^m$ such that $\lambda_0 := \|\nabla g(y^0;\underline{t})\|_2 \leq \frac{3\bar{c}_A}{4}$.
3. Set $\theta_0 := 1$ and $v^0 := y^0$.

Iteration: For $k = 0, 1, \cdots$, perform the following steps:

Step 1: If $\lambda_k \le \varepsilon$ then terminate.
Step 2: Compute $r^k := \theta_k^{-1}[v^k - (1-\theta_k)y^k]$.
Step 3: Update y^{k+1} as:

$$y^{k+1} := v^k - \alpha_k \nabla g(v^k, \underline{t}),$$

where $\alpha_k = \frac{\underline{t}}{\bar{c}_A(\bar{c}_A + \lambda_k)}$.
Step 4: Update $\theta_{k+1} := \frac{1}{2}\theta_k[(\theta_k^2 + 4)^{1/2} - \theta_k]$.
Step 5: Update

$$v^{k+1} := (1-\theta_{k+1})y^{k+1} + \theta_{k+1}(r^k - \rho_k \nabla g(v^k)),$$

where $\rho_k := \frac{\underline{t}}{2\bar{c}_A^2\theta_k}$.
Step 6: Solve (9) *in parallel* to obtain $x^*_{k+1} := x^*(v^{k+1}; \underline{t})$. Then form a gradient vector $\nabla g(v^{k+1}; \underline{t}) := Ax^*_{k+1} - b$ and compute $\lambda_{k+1} := \|\nabla g(v^{k+1}; \underline{t})\|_2$.

End.

The core step of Algorithm 2 is Step 6, where we need to solve M primal subproblems in parallel. Algorithm 2 differs from Nesterov's fast gradient algorithm [13] at Step 5, where v_{k+1} not only depends on y^{k+1} and v^k but also on $\nabla g(v^k; \underline{t})$.

The following theorem shows the convergence of Algorithm 2.

Theorem 2 *Let $y^0 \in \mathbb{R}^m$ be an initial point of Algorithm 2 such that $\|\nabla g(y^0; \underline{t})\|_2 \le \frac{3\bar{c}_A}{4}$. Then the sequence $\{(y^k, v^k)\}_{k\ge 0}$ generated by Algorithm 2 satisfies:*

$$g(y^k; \underline{t}) - \underline{g}^*(\underline{t}) \le \frac{4\bar{c}_A^2}{\underline{t}(k+1)^2}\|y^0 - y^*(\underline{t})\|^2. \tag{49}$$

Proof From (42) and the update rule of θ_k, we have:

$$\begin{aligned}\theta_k^{-2}(g(y^{k+1}; \underline{t}) - \underline{g}^*) + \bar{c}_A^2\underline{t}^{-1}\|r^{k+1} - \underline{y}^*\|_2^2 &\le \theta_k^{-2}(1-\theta_k)(g(y^k; \underline{t}) - \underline{g}^*) + \bar{c}_A^2\underline{t}^{-1}\|r^k - \underline{y}^*\|_2^2\\ &\le \theta_{k-1}^{-2}(g(y^k; \underline{t}) - \underline{g}^*) + \bar{c}_A^2\underline{t}^{-1}\|r^k - \underline{y}^*\|_2^2\end{aligned}$$

By induction, we obtain from this inequality that:

$$\begin{aligned}\theta_{k-1}^{-2}(g(y^k; \underline{t}) - \underline{g}^*) &\le \theta_0^{-2}(g(y^1; \underline{t}) - \underline{g}^*) + \bar{c}_A^2\underline{t}^{-1}\|r^1 - \underline{y}^*\|_2^2\\ &\le (1-\theta_0)\theta_0^{-2}(g(y^0; \underline{t}) - \underline{g}^*) + \bar{c}_A^2\underline{t}^{-1}\|r^0 - \underline{y}^*\|_2^2,\end{aligned}$$

for $k \ge 1$. Since $\theta_0 = 1$ and $y^0 = v^0$, we have $r^0 = y^0$ and the last inequality implies $g(y^k; \underline{t}) - \underline{g}^* \le \bar{c}_A^2\theta_{k-1}^2\underline{t}^{-1}\|y^0 - \underline{y}\|_2^2$. Since $\theta_{k-1} \le \frac{2}{k+1}$ due to Lemma 9, we obtain (49). □

Let us denote by:

$$\mathscr{R}(\bar{c}_A; \underline{t}) := \left\{ y^0 \in \mathbb{R}^m \mid \|\nabla g(y^0; \underline{t})\|_2 \le \frac{3\bar{c}_A}{4} \right\}. \tag{50}$$

It is obvious that $y^*(\underline{t}) \in \mathscr{R}(\bar{c}_A; \underline{t})$. This set is a neighbourhood of the solution $y^*(\underline{t})$ of the problem (16).

Remark 4 Let $\varepsilon > 0$ be a given accuracy. If we fix the barrier parameter $\underline{t} := \varepsilon$ then the worst-case complexity of Algorithm 2 in the neighbourhood $\mathscr{R}(\bar{c}_A; \underline{t})$ is $O(\frac{2\bar{c}_A\underline{r}_0}{\varepsilon})$, where $\underline{r}_0 := r_0(\underline{t})$.

Remark 5 (*Switching strategy*) We can combine Algorithms 1 and 2 to obtain a switching variant:

– First, we apply Algorithm 1 to find a point $\hat{y}^0 \in \mathbb{R}^m$ and $\underline{t} > 0$ such that $\|\nabla g(\hat{y}^0;\underline{t})\|_2 \leq \frac{3\bar{c}_A}{4}$.
– Then, we switch to use Algorithm 2.

We can also replace the constant $\bar{c}_A$ in Algorithm 2 by any upper bound $\hat{c}_A$ of $\underline{c}_k$. For instance, we can choose $\hat{c}_A := \max\left\{\bar{c}_A, 4\|\nabla g(y^0;\underline{t})\|_2/3\right\}$.

6 Numerical tests

In this section, we test the switching variant of Algorithms 1 and 2 proposed in Remark 5 which we name by PFGDA for solving the following convex programming problem:

$$\begin{array}{ll}\min\limits_{x\in\mathbb{R}^n} & \gamma\|x\|_1 + f(x)\\ \text{s.t.} & Ax = b,\ l \leq x \leq u,\end{array} \tag{51}$$

where $f(x) := \sum_{i=1}^n f_i(x_i)$, and $f_i : \mathbb{R} \to \mathbb{R}$ is a convex function, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $l, u \in \mathbb{R}^n$ such that $l \leq 0 < u$.

We note that the feasible set $X := [l,u]$ can be decomposed into n intervals $X_i := [l_i, u_i]$ and each interval is endowed with a 2-self concordant barrier $F_i(x_i) := -\ln(x_i - l_i) - \ln(u_i - x_i) + 2\ln((u_i - l_i)/2)$ for $i = 1,\dots,n$. Moreover, if we define $\phi(x) := -\sum_{i=1}^n [f_i(x_i) + \gamma|x_i|]$ then ϕ is concave and separable. Problem (51) can be reformulated equivalently to (SepCP).

The smoothed dual function components $g_i(y;t)$ of (51) can be written as:

$$g_i(y;t) = \max_{l_i<x_i<u_i}\left\{-f_i(x_i) - \gamma|x_i| + (A_i^T y)x_i - tF_i(x_i)\right\} - b^T y/n,$$

for $i = 1,\dots,n$. This one-variable minimization problem is nonsmooth but it can be solved easily. In particular, if f_i is affine then this problem can be solved in a *closed form*. In case f_i is smooth, we can reformulate (51) into a smooth convex program by adding n slack variables and $2n$ additional inequality constraints to handle the $\|x\|_1$ part.

We have implemented PFGDA in C++ running on a 16 cores Intel ®Xeon 2.7GHz workstation with 12 GB of RAM. The algorithm was parallelized by using OpenMP. We terminated PFGDA if:

$$\texttt{optim} := \|\nabla g(y^k;t_k)\|_2/\max\left\{1, \|\nabla g(y^0;t_0)\|_2\right\} \leq 10^{-3} \text{ and } t_k \leq 10^{-2}.$$

We have also implemented two algorithms, namely *decomposition algorithm with two primal steps* [20, Algorithm 1] and *decomposition algorithm with two dual steps* in [19, Algorithm 1] which we named 2pDecompAlg and 2dDecompAlg, respectively, for solving problem (51) and compared them with PFGDA. We terminated 2pDecompAlg and 2dDecompAlg by using the same conditions as in [19,20] with the tolerances $\varepsilon_{\text{feas}} = \varepsilon_{\text{fun}} = \varepsilon_{\text{obj}} = 10^{-3}$ and $j_{\max} = 3$. We also terminated all three algorithms if the maximum number of iterations maxiter $:= 10{,}000$ was reached. In the last case we clarify that the algorithm is failed.

a. Basis pursuit problem. If the function $f(x) \equiv 0$ for all x then problem (51) becomes a bound constrained basis pursuit problem to recover the sparse coefficient vector x of given signals based on a transform operator A and a vector of observations b. We assume that $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$, where $m < n$ and x has k nonzero elements ($k \ll n$).

In this case, we only illustrate PFGDA by applying it to solve some small size test problems. In order to generate a test problem, we generate a random orthogonal matrix A and a random vector x_0 which has k nonzero elements. Then we define vector b as $b := Ax_0$.

We test PFGDA on the four problems such that $[m,n,k]$ are $[50,128,14]$, $[100,256,20]$, $[200,512,30]$ and $[500,1024,50]$. The results reported by PFGDA are plotted in Figure 1.

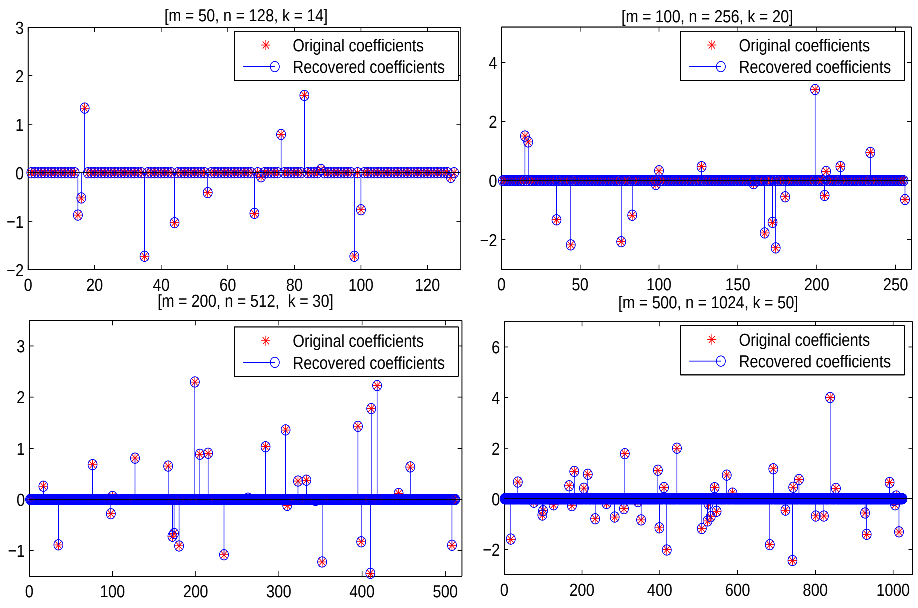

Fig. 1 Illustration of PFGDA via the basis pursuit problem

As we can see from these figures that the vector of recovered coefficients x matches very well the vector of original coefficients x_0 in these four problems. Moreover, PFGDA requires $376, 334, 297$ and 332 iterations, respectively in the four problems.

b. Nonlinear separable convex problems. In order to test the performance of PFGDA, we generate in this case a large test-set of problems and compare the performance of PFGDA with 2pDecompAlg and 2dDecompAlg.

The test problems were generated as follows. We chose the objective function $f_i(x_i) := e^{-\gamma_i x_i} - 1$, where $\gamma_i > 0$ is a given parameter for $i = 1,\dots,n$. Matrix A was generated randomly in $[-1,1]$ and then was normalized by $A/\|A\|_\infty$. We generated a sparse vector x_0 randomly in $[-2,2]$ with the density 2% and defined a vector $b := A\bar{x}$. Vector $\gamma := (\gamma_1,\cdots,\gamma_n)^T$ was sparse and generated randomly in $[0,0.5]$. The lower bound l_i and the upper bounds u_i were set to -3 and 3, respectively for all $i = 1,\dots,n$.

We benchmarked three algorithms with performance profiles [4]. Recall that a performance profile is built based on a set $\mathscr{S}$ of n_s algorithms (solvers) and a collection $\mathscr{P}$ of n_p problems. Suppose that we build a profile based on computational time. We denote by $T_{p,s} :=$ *computational time required to solve problem p by solver s*. We compare the performance of algorithm s on problem p with the best performance of any algorithm on this problem; that is we compute the performance ratio $r_{p,s} := \frac{T_{p,s}}{\min\{T_{p,\hat{s}} \mid \hat{s}\in\mathscr{S}\}}$. Now, let $\rho_s(\tau) := \frac{1}{n_p}\mathrm{size}\left\{p\in\mathscr{P} \mid r_{p,s}\le\tau\right\}$ for $\tau\in\mathbb{R}_+$. The function $\rho_s : \mathbb{R}\to[0,1]$ is the probability for solver s that a performance ratio is within a factor τ of the best possible ratio. We use the term “performance profile” for the distribution function ρ_s of a performance metric. We plotted the performance profiles in log-scale, i.e. $\rho_s(\tau) := \frac{1}{n_p}\text{size}\left\{p \in \mathscr{P} \mid \log_2(r_{p,s}) \leq \log_2 \tau\right\}$.

We tested three algorithms on a collection of 50 random problems with m from 200 to $1,500$ and n from $1,000$ to $15,000$. The profiles are plotted in Figure 2. Based on this test, we

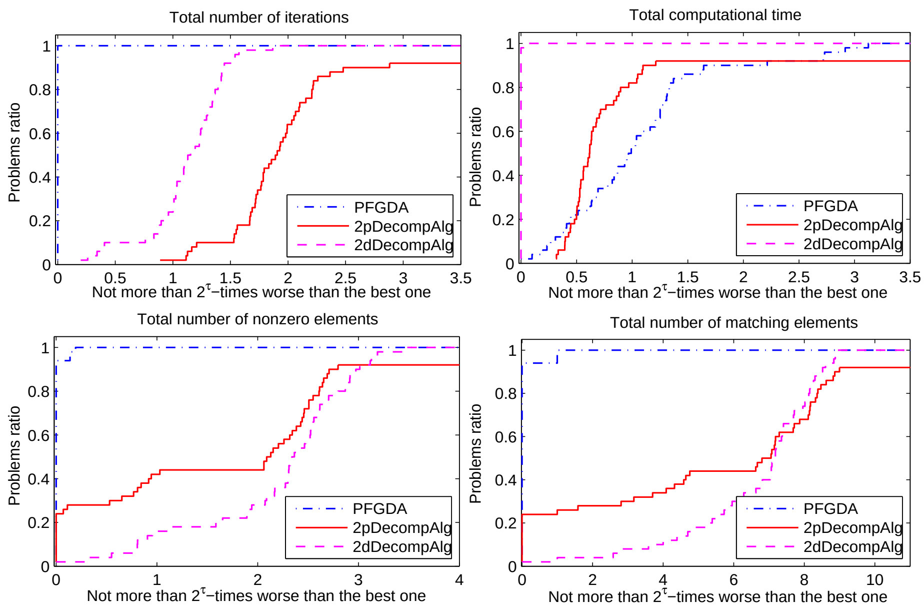

Fig. 2 Performance profiles in $\log_2$ scale of three algorithms.

can make the following observations. Both algorithms, `PDGDA` and `2dDecompAlg`, can solve all the test problems, while `2pDecompAlg` can only solve 46/50 (92%) problems. `PFGDA` requires a significantly fewer iterations than `2pDecompAlg` and `2dDecompAlg`, and it has the best performance on 100% problems in terms of number of iterations. `2dDecompAlg` is the best in terms of computational time where it reaches 100% the test problem with the best performance. However, the number of nonzero elements of the obtained solution in `PFGDA` matches very well the vector of original coefficients x_0, while it is rather bad in `2pDecompAlg` and `2dDecompAlg` as we can see from the last figure. In other words, `2dDecompAlg` is not good at finding a sparse solution in this example.

7 Concluding remarks

In this paper we have proposed two new dual gradient-based decomposition algorithms for solving large-scale separable convex optimization problems. We have analyzed the convergence of these to schemes and derived the rate of convergence. The first property of these methods is that they can handle general convex objective functions. Therefore, they can be applied to a wide range of applications compared to second order methods. Second, the new algorithms can implemented in parallel and all the algorithmic parameters are updated automatically without using any tuning strategy.Third, the convergence rate of Algorithm 2 is $O(1/k)$ which is optimal in the dual decomposition framework. Finally, the complexity estimates of the algorithms do not depend on the diameter of the feasible set as in proximal-type methods, they only depend on the parameter of the barrier functions.

Acknowledgements This research was supported by Research Council KUL: PFV/10/002 Optimization in Engineering Center OPTEC, GOA/10/09 MaNet; Flemish Government: IOF/KP/SCORES4CHEM, FWO: PhD/postdoc grants and projects: G.0320.08 (convex MPC), G.0377.09 (Mechatronics MPC); IWT: PhD Grants, projects: SBO LeCoPro; Belgian Federal Science Policy Office: IUAP P7 (DYSCO, Dynamical systems, control and optimization, 2012-2017); EU: FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), Eurostars SMART, ACCM; the European Union, Seventh Framework Programme (FP7/2007–2013), EMBOCON, under grant agreement no 248940; CNCS-UEFISCDI (project TE, no. 19/11.08.2010); ANCS (project PN II, no. 80EU/2010); Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labor, Family and Social Protection through the Financial Agreements POSDRU/89/1.5/S/62557.

References

1. Bertsekas, D., Tsitsiklis, J.N.: Parallel and distributed computation: Numerical methods. Prentice Hall (1989).
2. Boyd, S., Parikh, N., Chu, E., Peleato, B.: Distributed optimization and statistics via alternating direction method of multipliers. Foundations and Trends in Machine Learning **3**(1), 1–122 (2011).
3. Chen, G., Teboulle, M.: A proximal-based decomposition method for convex minimization problems. Math. Program. **64**, 81–101 (1994).
4. Dolan, E., Moré, J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002).
5. Duchi, J., Agarwal, A., Wainwright, M.: Dual averaging for distributed optimization: Convergence analysis and network scaling. IEEE Trans. Automatic Control **57**(3), 592–606 (2012).
6. Hamdi, A.: Two-level primal-dual proximal decomposition technique to solve large-scale optimization problems. Appl. Math. Comput. **160**, 921–938 (2005).
7. He, B., Tao, M., Xu, M., Yuan, X.: Alternating directions based contraction method for generally separable linearly constrained convex programming problems. Optimization *(to appear)* (2011).
8. Kojima, M., Megiddo, N., Mizuno, S., et al: Horizontal and vertical decomposition in interior point methods for linear programs. Technical report., Information Sciences, Tokyo Institute of Technology, Tokyo (1993).
9. Necoara, I., Savorgnan, C., Tran-Dinh, Q., Suykens, J.A.K., Diehl, M.: Distributed Nonlinear Optimal Control Using Sequential Convex Programming and Smoothing Techniques. In: Proceedings of the 48th IEEE Conference on Decision and Control. Shanghai, China (2009).
10. Necoara, I., Suykens, J.: Applications of a smoothing technique to decomposition in convex optimization. IEEE Trans. Automatic control **53**(11), 2674–2679 (2008).
11. Necoara, I., Suykens, J.: Interior-point lagrangian decomposition method for separable convex optimization. J. Optim. Theory and Appl. **143**(3), 567–588 (2009).
12. Nedíc, A., Ozdaglar, A.: Distributed subgradient methods for multi-agent optimization. IEEE Trans. Automatic Control **54**, 48–61 (2009).
13. Nesterov, Y.: Introductory lectures on convex optimization: a basic course, *Applied Optimization*, vol. 87. Kluwer Academic Publishers (2004).
14. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005).
15. Nesterov, Y.: Barrier subgradient method. Math. Program., Ser. B **127**, 31–56 (2011).
16. Nesterov, Y., Nemirovski, A.: Interior-point Polynomial Algorithms in Convex Programming. Society for Industrial Mathematics (1994)
17. Palomar, D., Chiang, M.: A Tutorial on Decomposition Methods for Network Utility Maximization. IEEE J. Selected Areas in Communications **24**(8), 1439–1451 (2006).
18. Ruszczyński, A.: On convergence of an augmented Lagrangian decomposition method for sparse convex optimization. *Math. Oper. Res.* **20**, 634–656 (1995).
19. Tran-Dinh, Q., Necoara, I., Savorgnan, C., Diehl, M.: An Inexact Perturbed Path-Following Method for Lagrangian Decomposition in Large-Scale Separable Convex Optimization. *SIAM J. Optim.* (under revision) (2012).
20. Tran-Dinh, Q., Savorgnan, C., Diehl, M.: Combining lagrangian decomposition and excessive gap smoothing technique for solving large-scale separable convex optimization problems. *Comput. Optim. Appl.* (under revision), 1–29 (2011). `http://arxiv.org/abs/1105.5427`.
21. Xiao, L., Johansson, M., Boyd, S.: Simultaneous routing and resource allocation via dual decomposition. IEEE Trans. Commun. **52**(7), 1136–1144 (2004).
22. Zhao, G.: A Lagrangian dual method with self-concordant barriers for multistage stochastic convex programming. Math. Progam. **102**, 1–24 (2005).